What is a triangle?

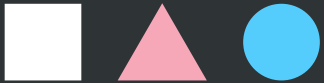
Astra Kolomatskaia

jww Mike Shulman

1

Introduction

Kindergarten Shapes



cube simplex globe

Semi-Simplicial Types

First, a type A_0 of *points*

$$A_0 : \mathsf{Type}$$

Semi-Simplicial Types

First, a type A_0 of *points*

$$A_0 : \mathsf{Type}$$

Second, for every two points x, $y : A_0$, a type $A_1\ x\ y$ of *lines* joining x and y

$$A_1 : (x : A_0)\ (y : A_0) \to \mathsf{Type}$$

Semi-Simplicial Types

First, a type A_0 of *points*

$$A_0 : \mathsf{Type}$$

Second, for every two points $x, y : A_0$, a type $A_1\ x\ y$ of *lines* joining x and y

$$A_1 : (x : A_0)\ (y : A_0) \to \mathsf{Type}$$

Third, for every three points $x, y, z : A_0$ and three lines $\alpha : A_1\ x\ y$, $\beta : A_1\ x\ z$, and $\gamma : A_1\ y\ z$, a type $A_2\ x\ y\ \alpha\ z\ \beta\ \gamma$ of *triangles* with the given boundary

$$A_2 : (x : A_0)\ (y : A_0)\ (\alpha : A_1\ x\ y)\ (z : A_0)\ (\beta : A_1\ x\ z)\ (\gamma : A_1\ y\ z) \to \mathsf{Type}$$

Semi-Simplicial Types *[cont.]*

A SST consists of an infinite list of the data that starts off as follows:

$A_0 : \mathsf{Type}$

Semi-Simplicial Types *[cont.]*

A SST consists of an infinite list of the data that starts off as follows:

$$A_0 : \mathsf{Type}$$

$$A_1 : (x_{01} : A_0)\ (x_{10} : A_0) \to \mathsf{Type}$$

Semi-Simplicial Types *[cont.]*

A SST consists of an infinite list of the data that starts off as follows:

$$A_0 : \mathsf{Type}$$

$$A_1 : (x_{01} : A_0)\ (x_{10} : A_0) \to \mathsf{Type}$$

$$A_2 : (x_{001} : A_0)\ (x_{010} : A_0)\ (\beta_{011} : A_1\ x_{001}\ x_{010})\ (x_{100} : A_0)\ (\beta_{101} : A_1\ x_{001}\ x_{100})$$
$$(\beta_{110} : A_1\ x_{010}\ x_{100}) \to \mathsf{Type}$$

Semi-Simplicial Types *[cont.]*

A SST consists of an infinite list of the data that starts off as follows:

$$A_0 : \mathsf{Type}$$

$$A_1 : (x_{01} : A_0)\ (x_{10} : A_0) \to \mathsf{Type}$$

$$A_2 : (x_{001} : A_0)\ (x_{010} : A_0)\ (\beta_{011} : A_1\ x_{001}\ x_{010})\ (x_{100} : A_0)\ (\beta_{101} : A_1\ x_{001}\ x_{100})\ (\beta_{110} : A_1\ x_{010}\ x_{100}) \to \mathsf{Type}$$

$$A_3 : (x_{0001} : A_0)\ (x_{0010} : A_0)\ (\beta_{0011} : A_1\ x_{0001}\ x_{0010})\ (x_{0100} : A_0)\ (\beta_{0101} : A_1\ x_{0001}\ x_{0100})\ (\beta_{0110} : A_1\ x_{0010}\ x_{0100})\ (\mathfrak{f}_{0111} : A_2\ x_{0001}\ x_{0010}\ \beta_{0011}\ x_{0100}\ \beta_{0101}\ \beta_{0110})\ (x_{1000} : A_0)\ (\beta_{1001} : A_1\ x_{0001}\ x_{1000})\ (\beta_{1010} : A_1\ x_{0010}\ x_{1000})\ (\mathfrak{f}_{1011} : A_2\ x_{0001}\ x_{0010}\ \beta_{0011}\ x_{1000}\ \beta_{1001}\ \beta_{1010})\ (\beta_{1100} : A_1\ x_{0100}\ x_{1000})\ (\mathfrak{f}_{1101} : A_2\ x_{0001}\ x_{0100}\ \beta_{0101}\ x_{1000}\ \beta_{1001}\ \beta_{1100})\ (\mathfrak{f}_{1110} : A_2\ x_{0010}\ x_{0100}\ \beta_{0110}\ x_{1000}\ \beta_{1010}\ \beta_{1100}) \to \mathsf{Type}$$

Semi-Simplicial Types *[cont.]*

A SST consists of an infinite list of the data that starts off as follows:

$$A_0 : \mathsf{Type}$$

$$A_1 : (x_{01} : A_0)\ (x_{10} : A_0) \to \mathsf{Type}$$

$$\begin{aligned} A_2 : {} & (x_{001} : A_0)\ (x_{010} : A_0)\ (\beta_{011} : A_1\ x_{001}\ x_{010})\ (x_{100} : A_0)\ (\beta_{101} : A_1\ x_{001}\ x_{100}) \\ & (\beta_{110} : A_1\ x_{010}\ x_{100}) \to \mathsf{Type} \end{aligned}$$

$$\begin{aligned} A_3 : {} & (x_{0001} : A_0)\ (x_{0010} : A_0)\ (\beta_{0011} : A_1\ x_{0001}\ x_{0010})\ (x_{0100} : A_0)\ (\beta_{0101} : A_1\ x_{0001}\ x_{0100}) \\ & (\beta_{0110} : A_1\ x_{0010}\ x_{0100})\ (\mathfrak{f}_{0111} : A_2\ x_{0001}\ x_{0010}\ \beta_{0011}\ x_{0100}\ \beta_{0101}\ \beta_{0110})\ (x_{1000} : A_0) \\ & (\beta_{1001} : A_1\ x_{0001}\ x_{1000})\ (\beta_{1010} : A_1\ x_{0010}\ x_{1000})\ (\mathfrak{f}_{1011} : A_2\ x_{0001}\ x_{0010}\ \beta_{0011}\ x_{1000}\ \beta_{1001}\ \beta_{1010}) \\ & (\beta_{1100} : A_1\ x_{0100}\ x_{1000})\ (\mathfrak{f}_{1101} : A_2\ x_{0001}\ x_{0100}\ \beta_{0101}\ x_{1000}\ \beta_{1001}\ \beta_{1100}) \\ & (\mathfrak{f}_{1110} : A_2\ x_{0010}\ x_{0100}\ \beta_{0110}\ x_{1000}\ \beta_{1010}\ \beta_{1100}) \to \mathsf{Type} \end{aligned}$$

$$\ldots$$

The Problem

One of the biggest open problems in type theory was to define semi-simplicial types internal to a semantically general homotopy type theory

This is related to the problem of defining the hierarchy of points, lines, triangles, etc. valued in *spaces* internal to homotopy theory

The problem of defining *semi-simplicial sets* is easy, i.e. with h-sets instead of types [although the solutions available in Book HoTT are far from elegant]

Simplex Categories

Consider the category Δ whose objects are natural numbers

The number n represents a stack of $(n + 1)$ elements

Morphisms are order preserving injections:

Simplex Categories

Consider the category Δ whose objects are natural numbers

The number n represents a stack of $(n+1)$ elements

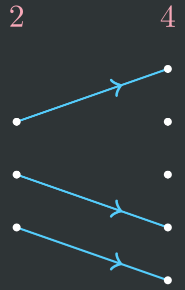
Morphisms are order preserving injections:

Simplex Categories

Consider the category Δ whose objects are natural numbers

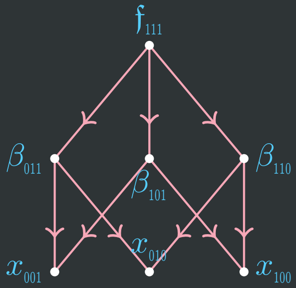
The number n represents a stack of $(n+1)$ elements

Morphisms are order preserving injections:



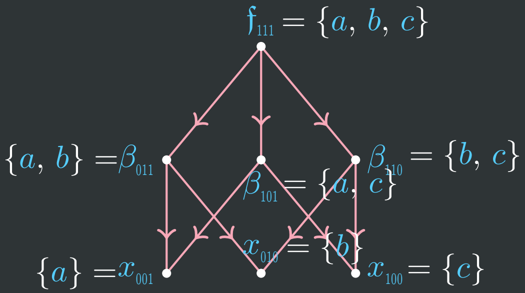
Simplex Categories *[cont.]*

Let A be a presheaf on Δ and consider $\mathfrak{f}_{111} : A_2$



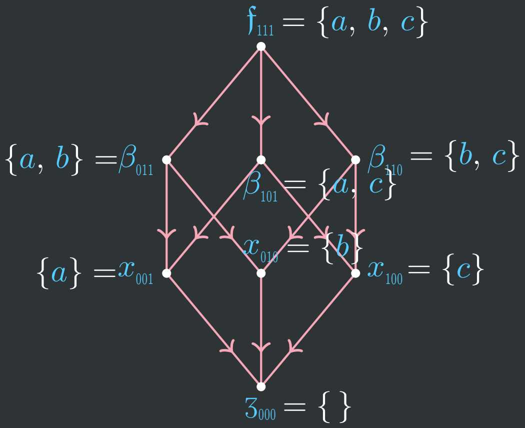
Simplex Categories *[cont.]*

Let A be a presheaf on Δ and consider $\mathfrak{f}_{111} : A_2$



Simplex Categories *[cont.]*

Let A be a presheaf on Δ and consider $\mathfrak{f}_{111} : A_2$



Simplex Categories *[cont.]*

We define Δ_+ to be the category of whole numbers starting from -1

The number -1 represents the stack with no elements

Δ_+ is obtained from Δ by adding an initial object

Simplex Categories *[cont.]*

We define Δ_+ to be the category of whole numbers starting from -1

The number -1 represents the stack with no elements

Δ_+ is obtained from Δ by adding an initial object

Δ is known as the *semi-simplex category*

Δ_+ is known as the *augmented semi-simplex category*

Semi-Simplicial Sets

A *semi-simplicial set* can be defined as a family of sets A_n for $n \geq 0$

Along with maps $\partial_k : A_n \to A_{n-1}$, for $k \in \{0, \dots, n\}$

$$
\begin{tikzcd}
A_0 & A_1 \arrow[l, shift left] \arrow[l, shift right] & A_2 \arrow[l, shift left=2] \arrow[l] \arrow[l, shift right=2] & A_3 \arrow[l, shift left=3] \arrow[l, shift left=1] \arrow[l, shift right=1] \arrow[l, shift right=3] & \dots
\end{tikzcd}
$$

These have to satisfy that:

$$\partial_k \circ \partial_l = \partial_{l-1} \circ \partial_k \quad \text{for } k < l$$

This is the *fibred* formulation

Semi-Simplicial Spaces

If we replace sets with spaces, then the condition on face maps is a homotopy:

$$\alpha_{k,l} : \partial_k \circ \partial_l \simeq \partial_{l-1} \circ \partial_k \quad \text{for } k < l$$

Semi-Simplicial Spaces

If we replace sets with spaces, then the condition on face maps is a homotopy:

$$\alpha_{k,l} : \partial_k \circ \partial_l \simeq \partial_{l-1} \circ \partial_k \quad \text{for } k < l$$

However, this is generally not satisfactory, as for $k < l < m$, we can prove that $\partial_k \circ \partial_l \circ \partial_m \simeq \partial_{m-2} \circ \partial_{l-1} \circ \partial_k$ in two different ways

Semi-Simplicial Spaces

If we replace sets with spaces, then the condition on face maps is a homotopy:

$$\alpha_{k,\,l} : \partial_k \circ \partial_l \simeq \partial_{l-1} \circ \partial_k \quad \text{for } k < l$$

However, this is generally not satisfactory, as for $k < l < m$, we can prove that $\partial_k \circ \partial_l \circ \partial_m \simeq \partial_{m-2} \circ \partial_{l-1} \circ \partial_k$ in two different ways

We require coherences $\beta_{k,\,l,\,m}$ that:

$$\beta_{k,\,l,\,m} : \alpha_{k,\,l} \star \partial_m \cdot \partial_{l-1} \star \alpha_{k,\,m} \cdot \alpha_{l-1,\,m-1} \star \partial_k \simeq \partial_k \star \alpha_{l,\,m} \cdot \alpha_{k,\,m-1} \star \partial_l \cdot \partial_{m-2} \star \alpha_{k,\,l}$$

Semi-Simplicial Spaces *[cont.]*

This last condition can be visualised as follows:

$$
\begin{tikzcd}
& \partial_{l-1} \circ \partial_k \circ \partial_m \arrow[r] \arrow[ddr, phantom, "\Downarrow"] & \partial_{l-1} \circ \partial_{m-1} \circ \partial_k \arrow[dr] & \\
\partial_k \circ \partial_l \circ \partial_m \arrow[ur] \arrow[dr] & & & \partial_{m-2} \circ \partial_{l-1} \circ \partial_k \\
& \partial_k \circ \partial_{m-1} \circ \partial_l \arrow[r] & \partial_{m-2} \circ \partial_k \circ \partial_l \arrow[ur] &
\end{tikzcd}
$$

Semi-Simplicial Spaces *[cont.]*

Next, considering a sequence of four consecutive face maps, we obtain a higher dimensional condition in the form of a *permutahedron*:

Writing down a formula for the required higher homotopy is very difficult!

Crux

When trying to construct SST in an indexed manner, one needs to prove a theorem about the construction

To prove that theorem, one needs to prove a *meta-theorem* about *the proof of the theorem*

To prove that meta-theorem, one needs to prove a *meta-meta-theorem* about *the proof of the meta-theorem*

...

And these theorems start to look a lot like the permutahedral coherences!

Overview

Mike Shulman and I constructed a new type theory called *Displayed Type Theory* that solves the problem of constructing SST

This solution is very satisfying because the idea behind it says something fundamentally new about semi-simplicial types, mathematically and independently of type theory

Overview *[cont.]*

To explain this solution, we first have to explain what it means to be a model of dependent type theory

Then, if we have any starting model, we build a new model, called the *simplicial model* in which *everything is a triangle*

Overview *[cont.]*

To explain this solution, we first have to explain what it means to be a model of dependent type theory

Then, if we have any starting model, we build a new model, called the *simplicial model* in which *everything is a triangle*

The key of this construction is that it is made in terms of *display*, which gives us new language only applicable in a diagram model

Using this new language, we can state a universal property for SST *as a diagram*, which enables a construction

Overview *[cont.]*

To explain this solution, we first have to explain what it means to be a model of dependent type theory

Then, if we have any starting model, we build a new model, called the *simplicial model* in which *everything is a triangle*

The key of this construction is that it is made in terms of *display*, which gives us new language only applicable in a diagram model

Using this new language, we can state a universal property for SST *as a diagram*, which enables a construction

Taking the discrete part of this diagram gives us an object SST in our arbitrary starting model

We thus have language for working with SST in full semantic generality

2

The Semantics of Dependent Type Theory

Settings for Space

There are various notions of *settings for homotopy theory*

These include model categories and ∞-toposes

Settings for Space

There are various notions of *settings for homotopy theory*

These include model categories and ∞-toposes

One key notion present in any such setting is that of a *fibration*

Settings for Space

There are various notions of *settings for homotopy theory*

These include model categories and ∞-toposes

One key notion present in any such setting is that of a *fibration*

The type theory notion of an indexed type semantically corresponds to a fibration

Categories with Families

In this talk, we will formally delve into one notion of fibrations

This is known as a *Category with Families (CwF)*

Categories with Families

In this talk, we will formally delve into one notion of fibrations

This is known as a *Category with Families (CwF)*

Definition: A CwF consists of a category $\mathcal{C}$, along with a chosen terminal object $\mathbb{1}$, and equipped with the data of two families of presheaves

$$\mathsf{Ty} : \mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$$

$$\mathsf{Tm} : \left(\int^{\mathcal{C}} \mathsf{Ty}\right)^{\mathsf{op}} \to \mathsf{Set}$$

Categories with Families

In this talk, we will formally delve into one notion of fibrations

This is known as a *Category with Families (CwF)*

Definition: A CwF consists of a category $\mathcal{C}$, along with a chosen terminal object $\mathbb{1}$, and equipped with the data of two families of presheaves

$$\mathsf{Ty} : \mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$$

$$\mathsf{Tm} : \left(\int^{\mathcal{C}} \mathsf{Ty}\right)^{\mathsf{op}} \to \mathsf{Set}$$

and, for every $\Gamma : \mathsf{ob}_{\mathcal{C}}$ and $A : \mathsf{Ty}\ \Gamma$, a chosen representation of the presheaf

$$\Delta \mapsto (\sigma : \mathsf{mor}_{\mathcal{C}}(\Delta, \Gamma)) \times \mathsf{Tm}\ \Delta\ A^{\sigma}$$

◁

Notation

The objects of the category $\mathcal{C}$ are called *contexts* and are denoted by Δ, Γ

For $\Gamma : \mathsf{ob}_{\mathcal{C}}$, we write

$$\Gamma \ \mathsf{ctx}$$

The *empty context* is the chosen terminal object $\mathbb{1}$, and is denoted by

$$(\,) \ \mathsf{ctx}$$

Notation

The objects of the category $\mathcal{C}$ are called *contexts* and are denoted by Δ, Γ

For $\Gamma : \mathsf{ob}_{\mathcal{C}}$, we write

$$\Gamma \ \mathsf{ctx}$$

The *empty context* is the chosen terminal object $\mathbb{1}$, and is denoted by

$$(\,) \ \mathsf{ctx}$$

The morphisms of $\mathcal{C}$ are called *substitutions* and are denoted by σ, τ

For $\sigma : \mathsf{mor}_{\mathcal{C}}(\Delta\ ,\ \Gamma)$, we write

$$\sigma : \Delta \to \Gamma$$

The unique substitution into the empty context is denoted by

$$[\,] : \Gamma \to (\,)$$

Notation *[cont.]*

The elements of the presheaf Ty are called *types* and are denoted by A, B

For $A : \mathsf{Ty}\ \Gamma$, we write

$$\gamma : \Gamma \vdash A\ \gamma\ \mathsf{type}$$

Notation *[cont.]*

The elements of the presheaf Ty are called *types* and are denoted by A, B

For $A : \mathsf{Ty}\ \Gamma$, we write

$$\gamma : \Gamma \vdash A\ \gamma\ \text{type}$$

The elements of the presheaf Tm are called *terms* and are denoted by t, s

For $t : \mathsf{Ty}\ \Gamma\ A$, we write

$$\gamma : \Gamma \vdash t\ \gamma : A\ \gamma$$

Notation *[cont.]*

The elements of the presheaf Ty are called *types* and are denoted by A, B

For $A : \mathsf{Ty}\ \Gamma$, we write

$$\gamma : \Gamma \vdash A\ \gamma\ \mathsf{type}$$

The elements of the presheaf Tm are called *terms* and are denoted by t, s

For $t : \mathsf{Ty}\ \Gamma\ A$, we write

$$\gamma : \Gamma \vdash t\ \gamma : A\ \gamma$$

We denote the functorial action of substitutions by

$$\frac{\sigma : \Delta \to \Gamma \qquad \gamma : \Gamma \vdash A\ \gamma\ \mathsf{type}}{\delta : \Delta \vdash A\ \ (\sigma\ \delta)\ \ \mathsf{type}} \qquad \frac{\sigma : \Delta \to \Gamma \qquad \gamma : \Gamma \vdash t\ \gamma : A\ \gamma}{\delta : \Delta \vdash t\ \ (\sigma\ \delta) : A\ \ (\sigma\ \delta)}$$

Notation *[cont.]*

We now consider the hypothesis of the chosen representation of

$$\Delta \mapsto (\sigma : \mathsf{mor}_{\mathcal{C}}(\Delta, \Gamma)) \times \mathsf{Tm}\ \Delta\ A^{\sigma}$$

Notation *[cont.]*

We now consider the hypothesis of the chosen representation of

$$\Delta \mapsto (\sigma : \mathsf{mor}_{\mathcal{C}}(\Delta, \Gamma)) \times \mathsf{Tm}\ \Delta\ A^{\sigma}$$

First, for Γ ctx and $\gamma : \Gamma \vdash A\ \gamma$ type, we get the representing object

$$(\gamma : \Gamma,\ a : A\ \gamma)\ \mathsf{ctx}$$

This is known as the *context extension*

Notation *[cont.]*

We then have a natural family of bijections:

$$\Big(\Delta \to (\gamma : \Gamma,\ a : A\ \gamma)\Big) \simeq \Big((\sigma : \Delta \to \Gamma) \times (\delta : \Delta \vdash t\ \delta : A\ \ (\sigma\ \delta))\Big)$$

Notation *[cont.]*

We then have a natural family of bijections:

$$\Big(\Delta \to (\gamma : \Gamma,\ a : A\ \gamma)\Big) \simeq \Big((\sigma : \Delta \to \Gamma) \times (\delta : \Delta \vdash t\ \delta : A\ \ (\sigma\ \delta))\Big)$$

By Yoneda, this is determined by evaluating at $1_{(\gamma : \Gamma,\ a : A\ \gamma)}$

Notation *[cont.]*

We then have a natural family of bijections:

$$\Big(\Delta \to (\gamma : \Gamma,\ a : A\ \gamma)\Big) \simeq \Big((\sigma : \Delta \to \Gamma) \times (\delta : \Delta \vdash t\ \delta : A\ (\sigma\ \delta))\Big)$$

By Yoneda, this is determined by evaluating at $1_{(\gamma : \Gamma,\ a : A\ \gamma)}$

The first component gives the *parent map*:

$$\mathsf{pt}^{A} : (\gamma : \Gamma,\ a : A\ \gamma) \to \Gamma$$

Notation *[cont.]*

We then have a natural family of bijections:

$$\Big(\Delta \to (\gamma : \Gamma,\ a : A\ \gamma)\Big) \simeq \Big((\sigma : \Delta \to \Gamma) \times (\delta : \Delta \vdash t\ \delta : A\ \ (\sigma\ \delta))\Big)$$

By Yoneda, this is determined by evaluating at $1_{(\gamma : \Gamma,\ a : A\ \gamma)}$

The first component gives the *parent map*:

$$\mathsf{pt}^{A} : (\gamma : \Gamma,\ a : A\ \gamma) \to \Gamma$$

The second component gives the *zero variable*:

$$\gamma : \Gamma,\ a : A\ \gamma \vdash \mathsf{zv}^{A}\ \ [\,\gamma,\ a\,] : A\ \ \big(\mathsf{pt}^{A}\ \ [\,\gamma,\ a\,]\big)$$

Notation *[cont.]*

Given the natural family of bijections:

$$\left(\Delta \to (\gamma : \Gamma,\ a : A\ \gamma)\right) \simeq \left((\sigma : \Delta \to \Gamma) \times (\delta : \Delta \vdash t\ \delta : A\ (\sigma\ \delta))\right)$$

By Yoneda, the forwards direction of the map is:

$$\left(\tau : \Delta \to (\gamma : \Gamma,\ a : A\ \gamma)\right) \mapsto \left(\mathsf{pt}^{A} \circ \tau,\ \left(\mathsf{zv}^{A}\right)^{\tau}\right)$$

Notation *[cont.]*

Given the natural family of bijections:

$$\Big(\Delta \to (\gamma : \Gamma,\ a : A\ \gamma)\Big) \simeq \Big((\sigma : \Delta \to \Gamma) \times (\delta : \Delta \vdash t\ \delta : A\ (\sigma\ \delta))\Big)$$

By Yoneda, the forwards direction of the map is:

$$\Big(\tau : \Delta \to (\gamma : \Gamma,\ a : A\ \gamma)\Big) \mapsto \Big(\mathsf{pt}^{A} \circ \tau,\ (\mathsf{zv}^{A})^{\tau}\Big)$$

We have a map in the reverse direction representing *substitution extension*:

$$\frac{\sigma : \Delta \to \Gamma \qquad \delta : \Delta \vdash t\ \delta : A\ (\sigma\ \delta)}{[\,\sigma,\ t\,] : \Delta \to (\gamma : \Gamma,\ a : A\ \gamma)}$$

Notation *[cont.]*

The fact that the two maps outlined above are inverse bijections says that:

$\mathrm{I}.$ For $\sigma : \Delta \to \Gamma$ and $\delta : \Delta \vdash t\ \delta : A\ (\sigma\ \delta)$,

$$\mathsf{pt}^A \circ [\,\sigma,\ t\,] \equiv \sigma$$
$$\left(\mathsf{zv}^A\right)^{[\,\sigma,\ t\,]} \equiv t$$

$\mathrm{II}.$ For $\tau : \Delta \to (\gamma : \Gamma,\ a : A\ \gamma)$,

$$\left[\ \mathsf{pt}^A \circ \tau,\ \left(\mathsf{zv}^A\right)^{\tau}\ \right] \equiv \tau$$

Notation *[cont.]*

Note that the construction above say that the following diagram is a pullback:

$$
\begin{tikzcd}
(\delta : \Delta,\ a : A\ (\sigma\ \delta)) \arrow[r, "{[\,\sigma \circ \mathsf{pt}^{A^\sigma},\ \mathsf{zv}^{A^\sigma}\,]}"] \arrow[d, "\mathsf{pt}^{A^\sigma}"'] \arrow[dr, phantom, very near start, "\lrcorner"] & (\gamma : \Gamma,\ a : A\ \gamma) \arrow[d, "\mathsf{pt}^{A}"] \\
\Delta \arrow[r, "\sigma"'] & \Gamma
\end{tikzcd}
$$

Notation *[cont.]*

Note that the construction above say that the following diagram is a pullback:

$$\begin{tikzcd}
(\delta : \Delta,\ a : A\ (\sigma\ \delta)) \arrow[r, "{[\sigma \circ \mathsf{pt}^{A^\sigma},\ \mathsf{zv}^{A^\sigma}]}"] \arrow[d, "\mathsf{pt}^{A^\sigma}"'] \arrow[dr, phantom, "\lrcorner", very near start] & (\gamma : \Gamma,\ a : A\ \gamma) \arrow[d, "\mathsf{pt}^{A}"] \\
\Delta \arrow[r, "\sigma"'] & \Gamma
\end{tikzcd}$$

Thus we have a strictly functorial assignment of distinguished pullbacks of parent maps along arbitrary substitutions

3

The Simplicial Model

Diagram Models

Let $\mathcal{C}$ be a category with families

We want to construct $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$, which is known as a *diagram model*

Diagram Models

Let $\mathcal{C}$ be a category with families

We want to construct $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$, which is known as a *diagram model*

The underlying category of $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$ is just $\mathcal{C}$-valued presheaves in Δ_+^{op}

The fibrant structure of $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$ is known as the *Reedy model structure*

Diagram Models

Let $\mathcal{C}$ be a category with families

We want to construct $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$, which is known as a *diagram model*

The underlying category of $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$ is just $\mathcal{C}$-valued presheaves in Δ_+^{op}

The fibrant structure of $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$ is known as the *Reedy model structure*

However, we will give a new custom construction of this CwF structure that uses special properties of Δ_+^{op}

Diagram Models *[cont.]*

We will refer to the starting model $\mathcal{C}$ as the *discrete model*, denoted dm

We refer to the diagram model $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$ as the *simplicial model*, denoted sm

Diagram Models *[cont.]*

We will refer to the starting model $\mathcal{C}$ as the *discrete model*, denoted dm

We refer to the diagram model $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$ as the *simplicial model*, denoted sm

For $n \geq -2$, let Δ_+^n denote the full subcategory of Δ_+ on objects $k \leq n$

In order to construct $\mathcal{C}^{\Delta_+^{\mathrm{op}}}$, we will first construct $\mathcal{C}^{\Delta_+^{n\,\mathrm{op}}}$

We will refer to $\mathcal{C}^{\Delta_+^{n\,\mathrm{op}}}$ as the *truncated simplicial model*, denoted sm^n

Categorical Structure

Recall that in Δ_+, the objects are whole numbers $\langle k \rangle$ for $k \geq -1$

Let $\mathbb{B}$ be the type of binary digits, which are $\mathbb{0}$, $\mathbb{1} : \mathbb{B}$

For $n \geq m \geq -1$, let $\mathbb{B}^{\langle n \rangle, \langle m \rangle}$ be the type of length $n + 1$ binary sequences such that exactly $m + 1$ of the digits have value $\mathbb{1}$

Categorical Structure

Recall that in Δ_+, the objects are whole numbers $\langle k \rangle$ for $k \geq -1$

Let $\mathbb{B}$ be the type of binary digits, which are $\mathbb{0}$, $\mathbb{1} : \mathbb{B}$

For $n \geq m \geq -1$, let $\mathbb{B}^{\langle n \rangle, \langle m \rangle}$ be the type of length $n + 1$ binary sequences such that exactly $m + 1$ of the digits have value $\mathbb{1}$

The identities $1_{\langle n \rangle}$ are given by length $n + 1$ sequences of the digit $\mathbb{1}$

For $b : \mathbb{B}^{\langle n \rangle, \langle k \rangle}$, we obtain $\mathbb{0}b : \mathbb{B}^{\langle n+1 \rangle, \langle k \rangle}$ and $\mathbb{1}b : \mathbb{B}^{\langle n+1 \rangle, \langle k+1 \rangle}$ by left appending

$$
\begin{aligned}
\mathbb{0}b_1 \circ b_0 &\equiv \mathbb{0}\,(b_1 \circ b_0) \\
\mathbb{1}b_1 \circ \mathbb{1}b_0 &\equiv \mathbb{1}\,(b_1 \circ b_0) \\
\mathbb{1}b_1 \circ \mathbb{0}b_0 &\equiv \mathbb{0}\,(b_1 \circ b_0)
\end{aligned}
$$

Categorical Structure *[cont.]*

If Γ $\mathsf{ctx}_{\mathsf{sm}^n}$, then Γ is a $\mathscr{C}$-valued presheaf on Δ^n_+

Thus we have, for $m \geq -2$, that Γ_m $\mathsf{ctx}_{\mathsf{dm}}$, such that $\Gamma_{-2} \equiv (\,)_{\mathsf{dm}}$

Categorical Structure *[cont.]*

If Γ $\mathsf{ctx}_{\mathsf{sm}^n}$, then Γ is a $\mathscr{C}$-valued presheaf on Δ_+^n

Thus we have, for $m \geq -2$, that Γ_m $\mathsf{ctx}_{\mathsf{dm}}$, such that $\Gamma_{-2} \equiv (\,)_{\mathsf{dm}}$

Also, for any $b : \mathbb{B}^{\langle n \rangle, \langle m \rangle}$, we have $\Gamma^b : \Gamma_n \to \Gamma_m$

We write γ^b for $\Gamma^b\ \gamma$

Categorical Structure *[cont.]*

There are two functors of relevance – *truncation* and *décalage*

$$\pi : \mathcal{C}^{\Delta_+^{n+1\,\mathrm{op}}} \to \mathcal{C}^{\Delta_+^{n\,\mathrm{op}}} \qquad (-)^{\mathsf{D}} : \mathcal{C}^{\Delta_+^{n+1\,\mathrm{op}}} \to \mathcal{C}^{\Delta_+^{n\,\mathrm{op}}}$$

$$(\pi\Gamma)_{m+1} \equiv \Gamma_{m+1} \qquad \left(\Gamma^{\mathsf{D}}\right)_{m+1} \equiv \Gamma_{m+2}$$

$$(\pi\Gamma)^{b} \equiv \Gamma^{b} \qquad \left(\Gamma^{\mathsf{D}}\right)^{b} \equiv \Gamma^{\mathbb{1}b}$$

$$(\pi\sigma)_{m+1} \equiv \sigma_{m+1} \qquad \left(\sigma^{\mathsf{D}}\right)_{m+1} \equiv \sigma_{m+2}$$

Categorical Structure *[cont.]*

There are two functors of relevance – *truncation* and *décalage*

$$\pi : \mathcal{C}^{\Delta_+^{n+1\,\mathrm{op}}} \to \mathcal{C}^{\Delta_+^{n\,\mathrm{op}}}$$
$$(\pi\Gamma)_{m+1} \equiv \Gamma_{m+1}$$
$$(\pi\Gamma)^b \equiv \Gamma^b$$
$$(\pi\sigma)_{m+1} \equiv \sigma_{m+1}$$

$$(-)^{\mathsf{D}} : \mathcal{C}^{\Delta_+^{n+1\,\mathrm{op}}} \to \mathcal{C}^{\Delta_+^{n\,\mathrm{op}}}$$
$$\left(\Gamma^{\mathsf{D}}\right)_{m+1} \equiv \Gamma_{m+2}$$
$$\left(\Gamma^{\mathsf{D}}\right)^b \equiv \Gamma^{\mathbb{1}b}$$
$$\left(\sigma^{\mathsf{D}}\right)_{m+1} \equiv \sigma_{m+2}$$

There is a natural transformation between them:

$$\rho : (-)^{\mathsf{D}} \Rightarrow \pi$$
$$(\rho_\Gamma)_{m+1} \equiv \Gamma^{\mathbb{0}1_{\langle m+1\rangle}}$$

Intuition

At the most basic level, we would like to define the judgement

$$\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} A\ \gamma\ \text{type}$$

Intuition

At the most basic level, we would like to define the judgement

$$\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} A\ \gamma\ \text{type}$$

A simplicial type consists of its discrete m-simplex types for $m \leq n+1$

$$\begin{aligned}
\gamma_{-1} : \Gamma_{-1} &\vdash_{\mathsf{dm}} A_{-1}\ \gamma_{-1}\ \text{type} \\
\gamma_0 : \Gamma_0,\ \mathfrak{z}_0 : A_{-1}\ \gamma_0{}^{\mathbb{0}} &\vdash_{\mathsf{dm}} A_0\ \gamma_0\ \mathfrak{z}_0\ \text{type} \\
\gamma_1 : \Gamma_1,\ \mathfrak{z}_{00} : A_{-1}\ \gamma_1{}^{\mathbb{00}},\ x_{01} : A_0\ \gamma_1{}^{\mathbb{01}}\ \mathfrak{z}_{00},\ x_{10} : A_0\ \gamma_1{}^{\mathbb{10}}\ \mathfrak{z}_{00} &\vdash_{\mathsf{dm}} A_1\ \gamma_1\ \mathfrak{z}_{00}\ x_{01}\ x_{10}\ \text{type} \\
&\vdots
\end{aligned}$$

Intuition

At the most basic level, we would like to define the judgement

$$\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} A\ \gamma\ \text{type}$$

A simplicial type consists of its discrete m-simplex types for $m \leq n+1$

$$\begin{aligned}
\gamma_{-1} : \Gamma_{-1} &\vdash_{\mathsf{dm}} A_{-1}\ \gamma_{-1}\ \text{type} \\
\gamma_0 : \Gamma_0,\ \mathfrak{z}_0 : A_{-1}\ \gamma_0{}^{\mathbb{0}} &\vdash_{\mathsf{dm}} A_0\ \gamma_0\ \mathfrak{z}_0\ \text{type} \\
\gamma_1 : \Gamma_1,\ \mathfrak{z}_{00} : A_{-1}\ \gamma_1{}^{\mathbb{00}},\ x_{01} : A_0\ \gamma_1{}^{\mathbb{01}}\ \mathfrak{z}_{00},\ x_{10} : A_0\ \gamma_1{}^{\mathbb{10}}\ \mathfrak{z}_{00} &\vdash_{\mathsf{dm}} A_1\ \gamma_1\ \mathfrak{z}_{00}\ x_{01}\ x_{10}\ \text{type} \\
&\vdots
\end{aligned}$$

We will write the type declarations of A_{n+1} generically as:

$$\gamma_{n+1} : \Gamma_{n+1},\ \partial a : \pi A_{\partial(n+1)}\ \gamma_{n+1} \vdash_{\mathsf{dm}} A_{n+1}\ \gamma_{n+1}\ \partial a\ \text{type}$$

Intuition *[cont.]*

Similarly, for terms, we would like to define the judgement

$$\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} t\ \gamma : A\ \gamma$$

Intuition *[cont.]*

Similarly, for terms, we would like to define the judgement

$$\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} t\ \gamma : A\ \gamma$$

A simplicial term consists of its discrete m-simplex components for $m \leq n+1$

$$\begin{aligned}
\gamma_{-1} : \Gamma_{-1} &\vdash_{\mathsf{dm}} t_{-1}\ \gamma_{-1} : A_{-1}\ \gamma_{-1} \\
\gamma_0 : \Gamma_0 &\vdash_{\mathsf{dm}} t_0\ \gamma_0 : A_0\ \gamma_0\ \left(t_{-1}\ {\gamma_0}^{\mathbb{0}}\right) \\
\gamma_1 : \Gamma_1 &\vdash_{\mathsf{dm}} t_1\ \gamma_1 : A_1\ \gamma_1\ \left(t_{-1}\ {\gamma_0}^{\mathbb{0}\mathbb{0}}\right)\ \left(t_0\ {\gamma_1}^{\mathbb{0}\mathbb{1}}\right)\ \left(t_0\ {\gamma_1}^{\mathbb{1}\mathbb{0}}\right) \\
&\vdots
\end{aligned}$$

Intuition *[cont.]*

Similarly, for terms, we would like to define the judgement

$$\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} t\ \gamma : A\ \gamma$$

A simplicial term consists of its discrete m-simplex components for $m \leq n+1$

$$\begin{aligned}
\gamma_{-1} : \Gamma_{-1} &\vdash_{\mathsf{dm}} t_{-1}\ \gamma_{-1} : A_{-1}\ \gamma_{-1} \\
\gamma_{0} : \Gamma_{0} &\vdash_{\mathsf{dm}} t_{0}\ \gamma_{0} : A_{0}\ \gamma_{0}\ \left(t_{-1}\ {\gamma_0}^{\mathbb{0}}\right) \\
\gamma_{1} : \Gamma_{1} &\vdash_{\mathsf{dm}} t_{1}\ \gamma_{1} : A_{1}\ \gamma_{1}\ \left(t_{-1}\ {\gamma_0}^{\mathbb{00}}\right)\ \left(t_{0}\ {\gamma_1}^{\mathbb{01}}\right)\ \left(t_{0}\ {\gamma_1}^{\mathbb{10}}\right) \\
&\vdots
\end{aligned}$$

We will write the type declarations of t_{n+1} generically as:

$$\gamma_{n+1} : \Gamma_{n+1} \vdash_{\mathsf{dm}} t_{n+1}\ \gamma_{n+1} : A_{n+1}\ \gamma_{n+1}\ \left(\pi t_{\partial(n+1)}\ \gamma_{n+1}\right)$$

Matching Objects

We now construct of the fibrant structure of the truncated simplicial model

As we saw, a key part of this is the *matching contexts* and *matching substitutions*

$$\frac{\gamma^- : \pi\Gamma \vdash_{\mathsf{sm}^n} A\ \gamma^-\ \text{type}}{\gamma_{n+1} : \Gamma_{n+1} \vdash_{\mathsf{dm}} A_{\partial(n+1)}\ \gamma_{n+1}\ \text{tel}}$$

$$\frac{\gamma^- : \pi\Gamma \vdash_{\mathsf{sm}^n} t\ \gamma^- : A\ \gamma^-}{\gamma_{n+1} : \Gamma_{n+1} \vdash_{\mathsf{dm}} t_{\partial(n+1)}\ \gamma_{n+1} : A_{\partial(n+1)}\ \gamma_{n+1}}$$

Types and Terms

Types and terms in the truncated simplicial model are then defined as follows:

$$\frac{\begin{array}{c}\gamma^{-} : \pi\Gamma \vdash_{\mathsf{sm}^{n}} \pi A\ \gamma^{-}\ \text{type} \\ \gamma_{n+1} : \Gamma_{n+1},\ \partial a : \pi A_{\partial(n+1)}\ \gamma_{n+1} \vdash_{\mathsf{dm}} A_{n+1}\ \gamma_{n+1}\ \partial a\ \text{type}\end{array}}{\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} A\ \gamma\ \text{type}}$$

$$\frac{\begin{array}{c}\gamma^{-} : \pi\Gamma \vdash_{\mathsf{sm}^{n}} \pi t\ \gamma^{-} : \pi A\ \gamma^{-} \\ \gamma_{n+1} : \Gamma_{n+1} \vdash_{\mathsf{dm}} t_{n+1}\ \gamma_{n+1} : A_{n+1}\ \gamma_{n+1}\ \left(\pi t_{\partial(n+1)}\ \gamma_{n+1}\right)\end{array}}{\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} t\ \gamma : A\ \gamma}$$

Extension

Extension of contexts by a type $\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} A\ \gamma$ type is obtained as follows

$$(\gamma : \Gamma,\ a : A\ \gamma)_{m+1} \equiv (\gamma^- : \pi\Gamma,\ a^- : \pi A\ \gamma^-)_{m+1} \quad \text{for} \quad m < n$$

$$(\gamma : \Gamma,\ a : A\ \gamma)_{n+1} \equiv (\gamma_{n+1} : \Gamma_{n+1},\ \partial a : \pi A_{\partial(n+1)}\ \gamma_{n+1},\ a : A_{n+1}\ \gamma_{n+1}\ \partial a)$$

Extension

Extension of contexts by a type $\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} A\ \gamma$ type is obtained as follows

$$\left(\gamma : \Gamma,\ a : A\ \gamma\right)_{m+1} \equiv \left(\gamma^- : \pi\Gamma,\ a^- : \pi A\ \gamma^-\right)_{m+1} \quad \text{for} \quad m < n$$
$$\left(\gamma : \Gamma,\ a : A\ \gamma\right)_{n+1} \equiv \left(\gamma_{n+1} : \Gamma_{n+1},\ \partial a : \pi A_{\partial(n+1)}\ \gamma_{n+1},\ a : A_{n+1}\ \gamma_{n+1}\ \partial a\right)$$

Extension of a substitution by a term $\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} t\ \gamma : A\ \gamma$ is obtained as follows

$$[\,\sigma,\ t\,]_{m+1} \equiv [\,\pi\sigma,\ \pi t\,]_{m+1} \quad \text{for} \quad m < n$$
$$[\,\sigma,\ t\,]_{n+1} \equiv \left[\,\sigma_{n+1},\ \pi t_{\partial(n+1)},\ t_{n+1}\,\right]$$

Display

Décalage comes with a fibrant counterpart known as *display*

$$\frac{\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} A\ \gamma\ \text{type}}{\gamma^{+} : \Gamma^{\mathsf{D}},\ a : \pi A^{\rho_\Gamma}\ \gamma^{+} \vdash_{\mathsf{sm}^{n}} A^{\mathsf{d}}\ \gamma^{+}\ a\ \text{type}} \qquad \frac{\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} t\ \gamma : A\ \gamma}{\gamma^{+} : \Gamma^{\mathsf{D}} \vdash_{\mathsf{sm}^{n}} t^{\mathsf{d}}\ \gamma^{+} : A^{\mathsf{d}}\ \gamma^{+}\ \pi t^{\rho_\Gamma}}$$

Display

Décalage comes with a fibrant counterpart known as *display*

$$\frac{\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} A\ \gamma \text{ type}}{\gamma^+ : \Gamma^{\mathsf{D}},\ a : \pi A^{\rho_\Gamma}\ \gamma^+ \vdash_{\mathsf{sm}^n} A^{\mathsf{d}}\ \gamma^+\ a \text{ type}} \qquad \frac{\gamma : \Gamma \vdash_{\mathsf{sm}^{n+1}} t\ \gamma : A\ \gamma}{\gamma^+ : \Gamma^{\mathsf{D}} \vdash_{\mathsf{sm}^n} t^{\mathsf{d}}\ \gamma^+ : A^{\mathsf{d}}\ \gamma^+\ \pi t^{\rho_\Gamma}}$$

Our construction will prove the following formulas for déclage:

$$(\gamma : \Gamma,\ a : A\ \gamma)^{\mathsf{D}} \equiv (\gamma^+ : \Gamma^{\mathsf{D}},\ a : \pi A^{\rho_\Gamma}\ \gamma^+,\ a' : A^{\mathsf{d}}\ \gamma^+\ a)$$
$$[\,\sigma,\ t\,]^{\mathsf{D}} \equiv [\,\sigma^{\mathsf{D}},\ \pi t^{\rho_\Delta},\ t^{\mathsf{d}}\,]$$

Inductive Definitions

Matching contexts and substitutions are inductively defined as follows:

$$
\begin{aligned}
A_{\partial(-1)} &\equiv (\,)_{\mathsf{dm}} \\
A_{\partial(n+2)}\ \gamma_{n+2} &\equiv \Big(\partial a : (\pi A^{\rho_\pi \Gamma})_{\partial(n+1)}\ \gamma_{n+2},\ a : (A^{\rho\Gamma})_{n+1}\ \gamma_{n+2}\ \partial a, \\
&\qquad\qquad \partial a' : (A^{\mathsf{d}})_{\partial(n+1)}\ [\,\gamma_{n+2},\ \partial a,\ a\,]\Big) \\
t_{\partial(-1)} &\equiv [\,]_{\mathsf{dm}} \\
t_{\partial(n+2)}\ \gamma_{n+2} &\equiv \Big[\,(\pi t^{\rho_\pi \Gamma})_{\partial(n+1)},\ (t^{\rho\Gamma})_{n+1},\ (t^{\mathsf{d}})_{\partial(n+1)}\Big]
\end{aligned}
$$

Inductive Definitions *[cont.]*

For display, we define:

$$\pi(A^{\mathsf{d}}) \equiv \pi A^{\mathsf{d}} \qquad \left(A^{\mathsf{d}}\right)_{n+1} \equiv A_{n+2}$$

$$\pi(t^{\mathsf{d}}) \equiv \pi t^{\mathsf{d}} \qquad \left(t^{\mathsf{d}}\right)_{n+1} \equiv t_{n+2}$$

Thus, just like décalage, display is a shift map

Inductive Definitions *[cont.]*

For display, we define:

$$\pi(A^{\mathsf{d}}) \equiv \pi A^{\mathsf{d}} \qquad \left(A^{\mathsf{d}}\right)_{n+1} \equiv A_{n+2}$$

$$\pi(t^{\mathsf{d}}) \equiv \pi t^{\mathsf{d}} \qquad \left(t^{\mathsf{d}}\right)_{n+1} \equiv t_{n+2}$$

Thus, just like décalage, display is a shift map

There is a lot more structure left to define a full CwF structure

What is presented here is sufficient to highlight the roles of décalage and display

Display in Type Theory

We can type theoretically present display as follows:

$$\Gamma \vdash_{\mathsf{sm}} A : \mathsf{Type} \qquad \Gamma \vdash_{\mathsf{sm}} t : A$$

$$\Gamma^{\mathsf{D}} \vdash_{\mathsf{sm}} A^{\mathsf{d}} : A^{\rho_\Gamma} \to \mathsf{Type} \qquad \Gamma^{\mathsf{D}} \vdash_{\mathsf{sm}} t^{\mathsf{d}} : A^{\mathsf{d}}\ t^{\rho_\Gamma}$$

$$\mathsf{Type}^{\mathsf{d}}\ A \equiv A \to \mathsf{Type} \qquad (\lambda\ x.\ t)^{\mathsf{d}} \equiv \lambda\ x\ x'.\ t^{\mathsf{d}}$$

$$(A \to B)^{\mathsf{d}}\ f \equiv (x : A) \to A^{\mathsf{d}}\ x \to B^{\mathsf{d}}\ (f\ x) \qquad (f\ a)^{\mathsf{d}} \equiv f^{\mathsf{d}}\ a\ a^{\mathsf{d}}$$

$$(\gamma : \Gamma,\ a : A)^{\mathsf{D}} \equiv (\gamma^{+} : \Gamma^{\mathsf{D}},\ a : A^{\rho_\Gamma},\ a' : A^{\mathsf{d}}\ a) \qquad (\,)^{\mathsf{D}}_{\mathsf{sm}} \equiv (\,)_{\mathsf{sm}}$$

$$x^{\mathsf{d}} \equiv x'$$

Parametricity

Consider the type of the polymorphic identity function:

$$\mathsf{T}_{\mathsf{id}} \equiv (A : \mathsf{Type}) \to A \to A$$

We calculate:

$$\mathsf{T}_{\mathsf{id}}{}^{\mathsf{d}}\ f \equiv (A : \mathsf{Type})\ (P : A \to \mathsf{Type})\ (x : A) \to P\,x \to P\ (f\ A\ x)$$

Then if $(\,) \vdash_{\mathsf{sm}} \mathsf{id} : \mathsf{T}_{\mathsf{id}}$, we have that id^{d} is a proof of this free theorem

Parametricity

Consider the type of the polymorphic identity function:

$$\mathsf{T}_{\mathsf{id}} \equiv (A : \mathsf{Type}) \to A \to A$$

We calculate:

$$\mathsf{T}_{\mathsf{id}}{}^{\mathsf{d}}\ f \equiv (A : \mathsf{Type})\ (P : A \to \mathsf{Type})\ (x : A) \to P\, x \to P\ (f\ A\ x)$$

Then if $(\,) \vdash_{\mathsf{sm}} \mathsf{id} : \mathsf{T}_{\mathsf{id}}$, we have that id^{d} is a proof of this free theorem

We then have:

$$\mathsf{idThm} \ : \ (\mathsf{id} \ :^{\triangle\Box} \ \mathsf{T}_{\mathsf{id}}) \ (A \ : \ \mathsf{Type}) \ (a \ : \ A) \ \to \ \mathsf{Path} \ A \ (\mathsf{id} \ A \ a) \ a$$
$$\mathsf{idThm} \ \mathsf{id} \ A \ a \ = \ \mathsf{id}^{\mathsf{d}} \ A \ (\lambda \ b \ \to \ \mathsf{Path} \ A \ b \ a) \ a \ \mathsf{refl}$$

Iterating Display

Now, what happens if we repeatedly apply $^{\mathsf{d}}$?

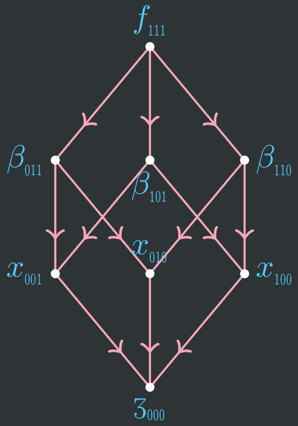
$$A : \mathsf{Type}$$

$$A^{\mathsf{d}} : A \to \mathsf{Type}$$

$$A^{\mathsf{dd}} : (\mathfrak{z}_{00} : A) \to A^{\mathsf{d}}\ \mathfrak{z}_{00} \to A^{\mathsf{d}}\ \mathfrak{z}_{00} \to \mathsf{Type}$$

$$\begin{aligned} A^{\mathsf{ddd}} : {} & (\mathfrak{z}_{000} : A)\ (x_{001} : A^{\mathsf{d}}\ \mathfrak{z}_{000})\ (x_{010} : A^{\mathsf{d}}\ \mathfrak{z}_{000}) \to \\ & A^{\mathsf{dd}}\ \mathfrak{z}_{000}\ x_{001}\ x_{010} \to (x_{100} : A^{\mathsf{d}}\ \mathfrak{z}_{000}) \to \\ & A^{\mathsf{dd}}\ \mathfrak{z}_{000}\ x_{001}\ x_{100} \to A^{\mathsf{dd}}\ \mathfrak{z}_{000}\ x_{010}\ x_{100} \to \mathsf{Type} \end{aligned}$$

Everything is a triangle!



4 Semi-Simplicial Types

Object Classifiers

In a CwF, an *object classifier* consists of:

- a *universe* type

$$\gamma : \Gamma \vdash \mathsf{Type}\,\gamma \text{ type}$$

Object Classifiers

In a CwF, an *object classifier* consists of:

i. a *universe* type

$$\gamma : \Gamma \vdash \mathsf{Type}\ \gamma\ \text{type}$$

ii. an *element* fibration

$$\gamma : \Gamma,\ A : \mathsf{Type}\ \gamma \vdash \mathsf{El}\ A\ \gamma\ \text{type}$$

Object Classifiers

In a CwF, an *object classifier* consists of:

i. a *universe* type

$$\gamma : \Gamma \vdash \mathsf{Type}\ \gamma\ \text{type}$$

ii. an *element* fibration

$$\gamma : \Gamma,\ A : \mathsf{Type}\ \gamma \vdash \mathsf{El}\ A\ \gamma\ \text{type}$$

iii. for every type $\gamma : \Gamma \vdash A\ \gamma$ type a *code* in the universe

$$\gamma : \Gamma \vdash \mathsf{Code}\ A\ \gamma : \mathsf{Type}\ \gamma$$

Object Classifiers

In a CwF, an *object classifier* consists of:

i a *universe* type

$$\gamma : \Gamma \vdash \mathsf{Type}\ \gamma\ \text{type}$$

ii an *element* fibration

$$\gamma : \Gamma,\ A : \mathsf{Type}\ \gamma \vdash \mathsf{El}\ A\ \gamma\ \text{type}$$

iii for every type $\gamma : \Gamma \vdash A\ \gamma$ type a *code* in the universe

$$\gamma : \Gamma \vdash \mathsf{Code}\ A\ \gamma : \mathsf{Type}\ \gamma$$

▷ such that the pullback of the El fibration along that code exactly yields the type A, that is

$$\gamma : \Gamma \vdash \mathsf{El}\ (\mathsf{Code}\ A)\ \gamma \equiv A\ \gamma$$

SSTs Homotopically

Now consider the problem of constructing a *classifier for semi-simplicial diagrams*

Such a classifier would consist of a type $\gamma : \Gamma \vdash \mathsf{SST}\ \gamma$ type, along with a simplicial diagram tower of the form

SSTs Homotopically

Now consider the problem of constructing a *classifier for semi-simplicial diagrams*

Such a classifier would consist of a type $\gamma : \Gamma \vdash \mathsf{SST}\ \gamma$ type, along with a simplicial diagram tower of the form

$$\gamma : \Gamma,\ A : \mathsf{SST}\ \gamma$$
$$\qquad \vdash \mathsf{El}_0\ A\ \text{type}$$
$$\gamma : \Gamma,\ A : \mathsf{SST}\ \gamma,\ a_{01} : \mathsf{El}_0\ \gamma\ A,\ a_{10} : \mathsf{El}_0\ \gamma\ A$$
$$\qquad \vdash \mathsf{El}_1\ \gamma\ A\ a_{01}\ a_{10}\ \text{type}$$
$$\gamma : \Gamma,\ A : \mathsf{SST}\ \gamma,\ a_{001} : \mathsf{El}_0\ \gamma\ A,\ a_{010} : \mathsf{El}_0\ \gamma\ A,\ a_{011} : \mathsf{El}_1\ \gamma\ A\ a_{001}\ a_{010}$$
$$\qquad a_{100} : \mathsf{El}_0\ \gamma\ A,\ a_{101} : \mathsf{El}_1\ \gamma\ A\ a_{100}\ a_{100},\ a_{110} : \mathsf{El}_1\ \gamma\ A\ a_{100}\ a_{010}$$
$$\qquad \vdash \mathsf{El}_2\ \gamma\ A\ a_{001}\ a_{010}\ a_{011}\ a_{100}\ a_{101}\ a_{110}\ \text{type}$$

$\ldots$

SSTs Homotopically *[cont.]*

This type SST and element fibrations El_n are such that for any simplicial diagram data over a context Γ, this data arises uniquely as the appropriate series of pullbacks constructed from some term $\gamma : \Gamma \vdash A\ \gamma : \mathsf{SST}\ \gamma$

This hypothesis would result in some SST analogue of Code

SSTs Homotopically *[cont.]*

This type SST and element fibrations El_n are such that for any simplicial diagram data over a context Γ, this data arises uniquely as the appropriate series of pullbacks constructed from some term $\gamma : \Gamma \vdash A\ \gamma : \mathsf{SST}\ \gamma$

This hypothesis would result in some SST analogue of Code

Stated in this way, this is an *infinitary* or *non-elementary* universal property

It refers to infinite diagrams indexed by the external set of natural numbers (as opposed to any internal natural-numbers object that may exist in $\mathcal{C}$)

SSTs Homotopically *[cont.]*

This type SST and element fibrations El_n are such that for any simplicial diagram data over a context Γ, this data arises uniquely as the appropriate series of pullbacks constructed from some term $\gamma : \Gamma \vdash A\ \gamma : \mathsf{SST}\ \gamma$

This hypothesis would result in some SST analogue of Code

Stated in this way, this is an *infinitary* or *non-elementary* universal property

It refers to infinite diagrams indexed by the external set of natural numbers (as opposed to any internal natural-numbers object that may exist in $\mathscr{C}$)

The problem of defining semi-simplicial types can roughly be thought of as one of giving a finitary universal property for such an object, so that it could be characterized and even constructed in a finitary syntactic type theory

Semi-Simplicial Types

Main Idea: *A semi-simplicial type X consists of a type X_0 together with, for every $x : X_0$, a displayed semi-simplicial type over X*

Semi-Simplicial Types

MAIN IDEA: *A semi-simplicial type X consists of a type X_0 together with, for every $x : X_0$, a displayed semi-simplicial type over X*

In Agda-esque syntax, we write this coinductive definition as:

$$
\begin{array}{l}
\textbf{codata}\ \mathsf{SST} : \mathsf{Type}\ \textbf{where} \\
\quad \mathsf{Z} : \mathsf{SST} \to \mathsf{Type} \\
\quad \mathsf{S} : (X : \mathsf{SST}) \to \mathsf{Z}\ X \to \mathsf{SST}^{\mathsf{d}}\ X
\end{array}
$$

Unfolding the Definition

$$\begin{array}{l}\textbf{codata}\ \mathsf{SST} : \mathsf{Type}\ \textbf{where}\\ \quad \mathsf{Z} : \mathsf{SST} \to \mathsf{Type}\\ \quad \mathsf{S} : (X : \mathsf{SST}) \to \mathsf{Z}\ X \to \mathsf{SST}^{\mathsf{d}}\ X\end{array}$$

Unfolding the Definition

$$\textbf{codata}\ \mathsf{SST} : \mathsf{Type}\ \textbf{where}$$
$$\mathsf{Z} : \mathsf{SST} \to \mathsf{Type}$$
$$\mathsf{S} : (X : \mathsf{SST}) \to \mathsf{Z}\ X \to \mathsf{SST}^{\mathsf{d}}\ X$$

Thus $A : \mathsf{SST}$ consists of:

i a type of 0-*simplicies*, $\mathsf{Z}\ A : \mathsf{Type}$

ii for every $x : \mathsf{Z}\ A$, a dependent SST called the *slice*, $\mathsf{S}\ A\ x : \mathsf{SST}^{\mathsf{d}}\ A$

Unfolding the Definition

$$\textbf{codata}\ \mathsf{SST} : \mathsf{Type}\ \textbf{where}$$
$$\mathsf{Z} : \mathsf{SST} \to \mathsf{Type}$$
$$\mathsf{S} : (X : \mathsf{SST}) \to \mathsf{Z}\ X \to \mathsf{SST}^{\mathsf{d}}\ X$$

Thus $A : \mathsf{SST}$ consists of:

i a type of 0-*simplicies*, $\mathsf{Z}\ A : \mathsf{Type}$

ii for every $x : \mathsf{Z}\ A$, a dependent SST called the *slice*, $\mathsf{S}\ A\ x : \mathsf{SST}^{\mathsf{d}}\ A$

Now if $B : \mathsf{SST}^{\mathsf{d}}\ A$, then B consists of:

i a family $\mathsf{Z}^{\mathsf{d}}\ B : \mathsf{Z}\ A \to \mathsf{Type}$

ii for every $x : \mathsf{Z}\ A$ and $x' : \mathsf{Z}^{\mathsf{d}}\ B\ x$, a doubly dependent SST, $\mathsf{S}^{\mathsf{d}}\ B\ x\ x' : \mathsf{SST}^{\mathsf{d}}\ A\ B\ (\mathsf{S}\ A\ x)$

Unfolding the Definition *[cont.]*

Given $A : \mathsf{SST}$, we get a type of zero-simplices by:

$$A_0 : \mathsf{Type}$$
$$A_0 \equiv \mathsf{Z}\ A$$

Unfolding the Definition *[cont.]*

Given $A : \mathsf{SST}$, we get a type of zero-simplices by:

$$A_0 : \mathsf{Type}$$
$$A_0 \equiv \mathsf{Z}\ A$$

Similarly, $B : \mathsf{SST}^{\mathsf{d}}\ A$, we get a type of dependent zero-simplices by:

$$B_0 : A_0 \to \mathsf{Type}$$
$$B_0\ y \equiv \mathsf{Z}^{\mathsf{d}}\ B\ y$$

Unfolding the Definition *[cont.]*

Given $A : \mathsf{SST}$, we get a type of zero-simplices by:

$$A_0 : \mathsf{Type}$$
$$A_0 \equiv \mathsf{Z}\ A$$

Similarly, $B : \mathsf{SST}^{\mathsf{d}}\ A$, we get a type of dependent zero-simplices by:

$$B_0 : A_0 \to \mathsf{Type}$$
$$B_0\ y \equiv \mathsf{Z}^{\mathsf{d}}\ B\ y$$

Putting this together, if we have two 0-simplices $x_{00}\ x_{10} : A_0$ of A, then we may form the type of 1-simplices of A as follows:

$$A_1 : (x_{01} : A_0)\ (x_{10} : A_0) \to \mathsf{Type}$$
$$A_1\ x_{01}\ x_{10} \equiv \mathsf{Z}^{\mathsf{d}}\ (\mathsf{S}\ A\ \ x_{01})\ x_{10},$$

Unfolding the Definition *[cont.]*

It therefore stands to reason that any $B : \mathsf{SST^d}\ A$ should have a type of dependent 1-simplices living over the 1-simplices of A

Thus if $\beta_{11} : A_1\ y_{01}\ y_{10}$, then given dependent endpoints $z_{01} : B_0\ y_{01}$ and $z_{10} : B_0\ y_{10}$, we should get a type $B_1\ y_{01}\ z_{01}\ y_{10}\ z_{10}\ \beta_{11}$, this is given by:

Unfolding the Definition *[cont.]*

It therefore stands to reason that any $B : \mathsf{SST}^{\mathsf{d}}\ A$ should have a type of dependent 1-simplices living over the 1-simplices of A

Thus if $\beta_{11} : A_1\ y_{01}\ y_{10}$, then given dependent endpoints $z_{01} : B_0\ y_{01}$ and $z_{10} : B_0\ y_{10}$, we should get a type $B_1\ y_{01}\ z_{01}\ y_{10}\ z_{10}\ \beta_{11}$, this is given by:

$$B_1 : (y_{01} : A_0)\ (z_{01} : B_0\ y_{01})\ (y_{10} : A_0)\ (z_{10} : B_0\ y_{10})\ (\beta_{11} : A_1\ y_{00}\ y_{10}) \to \mathsf{Type}$$
$$B_1\ y_{01}\ z_{01}\ y_{10}\ z_{10}\ \beta_{11} \equiv \mathsf{Z}^{\mathsf{dd}}\ (\mathsf{S}^{\mathsf{d}}\ B\ y_{01}\ z_{01})\ y_{10}\ z_{10}\ \beta_{11},$$

Unfolding the Definition *[cont.]*

Then, putting all of this together again, if we have a 0-simplex $x_{001} : A_0$, then we take $B \equiv \mathsf{S}\ A\ x_{00}$

For $x_{010} : A_0$, we have that $B_0\ x_{010} \equiv \mathsf{Z}^{\mathsf{d}}\ (\mathsf{S}\ A\ x_{001})\ x_{010} \equiv A_1\ x_{001}\ x_{010}$

Unfolding the Definition *[cont.]*

Then, putting all of this together again, if we have a 0-simplex $x_{001} : A_0$, then we take $B \equiv \mathsf{S}\ A\ x_{00}$

For $x_{010} : A_0$, we have that $B_0\ x_{010} \equiv \mathsf{Z}^{\mathsf{d}}\ (\mathsf{S}\ A\ x_{001})\ x_{010} \equiv A_1\ x_{001}\ x_{010}$

We thus get the type of 2-simplices of A as follows:

$$
\begin{aligned}
A_2 : (x_{001} : A_0)\ (x_{010} : A_0)\ (\beta_{011} : A_1\ x_{001}\ x_{010})\ (x_{100} : A_0) \\
(\beta_{101} : A_1\ x_{001}\ x_{100})\ (\beta_{110} : A_1\ x_{010}\ x_{100}) \to \mathsf{Type}
\end{aligned}
$$

$$
A_2\ x_{001}\ x_{010}\ \beta_{011}\ x_{100}\ \beta_{101}\ \beta_{110} \equiv \mathsf{Z}^{\mathsf{dd}}\ (\mathsf{S}^{\mathsf{d}}\ (\mathsf{S}\ A\ x_{001})\ x_{010}\ \beta_{011})\ x_{100}\ \beta_{101}\ \beta_{110}
$$

Unfolding the Definition *[cont.]*

Then, putting all of this together again, if we have a 0-simplex $x_{001} : A_0$, then we take $B \equiv \mathsf{S}\ A\ x_{00}$

For $x_{010} : A_0$, we have that $B_0\ x_{010} \equiv \mathsf{Z}^{\mathsf{d}}\ (\mathsf{S}\ A\ x_{001})\ x_{010} \equiv A_1\ x_{001}\ x_{010}$

We thus get the type of 2-simplices of A as follows:

$$\begin{aligned} A_2 :\ & (x_{001} : A_0)\ (x_{010} : A_0)\ (\beta_{011} : A_1\ x_{001}\ x_{010})\ (x_{100} : A_0) \\ & (\beta_{101} : A_1\ x_{001}\ x_{100})\ (\beta_{110} : A_1\ x_{010}\ x_{100}) \to \mathsf{Type} \\ A_2\ & x_{001}\ x_{010}\ \beta_{011}\ x_{100}\ \beta_{101}\ \beta_{110} \equiv \mathsf{Z}^{\mathsf{dd}}\ (\mathsf{S}^{\mathsf{d}}\ (\mathsf{S}\ A\ x_{001})\ x_{010}\ \beta_{011})\ x_{100}\ \beta_{101}\ \beta_{110} \end{aligned}$$

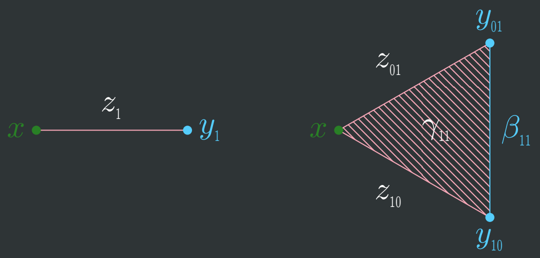
In general, this pattern continues in higher dimensions and the process described lets us extract n-simplex types.

Visualisation One

We can visualise what's going on in two different ways

The first visualisation shows how the n-simplices of the slice of A over x live dependently over simplices of A:

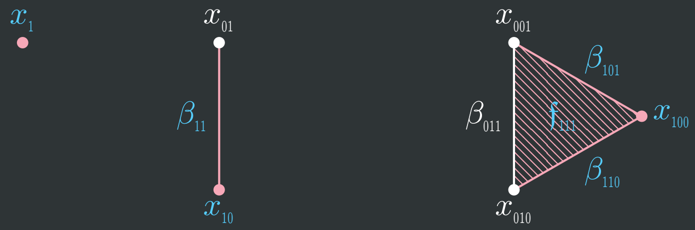
$$z_1 : (\mathsf{S}\ A\ x)_0\ \ y_1 \qquad \gamma_{11} : (\mathsf{S}\ A\ x)_1\ \ y_{01}\ \ z_{01}\ \ y_{10}\ \ z_{10}\ \ \beta_{11}$$



Visualisation Two

The second visualisation explains our formulas in terms of iterated slicing

$\mathsf{Z}\ A \qquad \mathsf{Z}^{\mathsf{d}}\ (\mathsf{S}\ A\ x_{01})\ x_{10} \qquad \mathsf{Z}^{\mathsf{dd}}\ (\mathsf{S}^{\mathsf{d}}\ (\mathsf{S}\ A\ x_{001})\ x_{010}\ \beta_{011})\ x_{100}\ \beta_{101}\ \beta_{110}$



The simplices of the slice are *mapping objects*

A Categorical Universal Property

We give the categorical universal property of SST in the simplicial mode

In general, there are issues of display modifying the context; here we will only give the UP in the empty context

A Categorical Universal Property

We give the categorical universal property of SST in the simplicial mode

In general, there are issues of display modifying the context; here we will only give the UP in the empty context

Suppose that Y closed type in the simplicial mode; we define an endofunctor by:

$$F\,(Y) \equiv \sum_{(v\,:\,Y)} \sum_{(A\,:\,\mathsf{Type})} (A \to Y^{\mathsf{d}}\ v)$$

This endofunctor comes with a copointing $\epsilon_Y : F\,(Y) \to Y$ by way of projection

A Categorical Universal Property

We give the categorical universal property of SST in the simplicial mode

In general, there are issues of display modifying the context; here we will only give the UP in the empty context

Suppose that Y closed type in the simplicial mode; we define an endofunctor by:

$$F\,(Y) \equiv \sum_{(\upsilon : Y)} \sum_{(A : \mathsf{Type})} (A \to Y^{\mathsf{d}}\ \upsilon)$$

This endofunctor comes with a copointing $\epsilon_Y : F\,(Y) \to Y$ by way of projection

Universal Property: Our characterization of SST is that it is a terminal coalgebra of the copointed endofunctor (F, ϵ)

A Categorical Universal Property *[cont.]*

$$F\,(Y) \equiv \sum_{(\upsilon\,:\,Y)} \sum_{(A\,:\,\mathsf{Type})} (A \to Y^{\mathsf{d}}\ \upsilon)$$

A Categorical Universal Property *[cont.]*

$$F(Y) \equiv \sum_{(\upsilon : Y)} \sum_{(A : \mathsf{Type})} (A \to Y^{\mathsf{d}}\ \upsilon)$$

UNIVERSAL PROPERTY: Our characterization of SST is that it is a terminal coalgebra of the copointed endofunctor (F, ϵ)

Thus, SST is the universal object equipped with a map

$$\mathsf{SST} \to \sum_{(X : \mathsf{SST})} \sum_{(A : \mathsf{Type})} (A \to \mathsf{SST}^{\mathsf{d}}\ X)$$

such that the first component of this map is the identity

A Categorical Universal Property *[cont.]*

$$F\,(Y) \equiv \sum_{(v\,:\,Y)} \sum_{(A\,:\,\mathsf{Type})} (A \to Y^{\mathsf{d}}\ v)$$

Universal Property: Our characterization of SST is that it is a terminal coalgebra of the copointed endofunctor (F, ϵ)

Thus, SST is the universal object equipped with a map

$$\mathsf{SST} \to \sum_{(X\,:\,\mathsf{SST})} \sum_{(A\,:\,\mathsf{Type})} (A \to \mathsf{SST}^{\mathsf{d}}\ X)$$

such that the first component of this map is the identity

What remains, therefore, is two components:

$$\mathsf{Z} : \mathsf{SST} \to \mathsf{Type}$$

$$\mathsf{S} : (X : \mathsf{SST}) \to \mathsf{Z}\ X \to \mathsf{SST}^{\mathsf{d}}\ X$$

A Categorical Universal Property *[cont.]*

What remains, therefore, is two components:

$$\mathsf{Z} : \mathsf{SST} \to \mathsf{Type}$$

$$\mathsf{S} : (X : \mathsf{SST}) \to \mathsf{Z}\ X \to \mathsf{SST}^{\mathsf{d}}\ X$$

A Categorical Universal Property *[cont.]*

What remains, therefore, is two components:

$$\mathsf{Z} : \mathsf{SST} \to \mathsf{Type}$$
$$\mathsf{S} : (X : \mathsf{SST}) \to \mathsf{Z}\ X \to \mathsf{SST}^{\mathsf{d}}\ X$$

This corresponds to our `Agda`-esque code:

```
codata SST : Type where
  Z : SST → Type
  S : (X : SST) → Z X → SSTᵈ X
```

Examples of SSTs

We can define several examples of SSTs

Examples of SSTs

We can define several examples of SSTs

First, we have the *singular semi-simplicial types*

$$\mathsf{Sing} \ : \ \mathsf{Type} \to \mathsf{SST}$$
$$\mathsf{Z} \ (\mathsf{Sing} \ A) = A$$
$$\mathsf{S} \ (\mathsf{Sing} \ A) \ x = \mathsf{Sing}^{\mathsf{d}} \ A \ (\lambda \ y \to \mathsf{Path} \ A \ x \ y)$$

Examples of SSTs

We can define several examples of SSTs

First, we have the *singular semi-simplicial types*

$$\mathsf{Sing} \;:\; \mathsf{Type} \to \mathsf{SST}$$
$$\mathsf{Z}\;(\mathsf{Sing}\;A) = A$$
$$\mathsf{S}\;(\mathsf{Sing}\;A)\;x = \mathsf{Sing}^{\mathsf{d}}\;A\;(\lambda\;y \to \mathsf{Path}\;A\;x\;y)$$

Next, we can define products of SSTs

$$\_\otimes\_ \;:\; \mathsf{SST} \to \mathsf{SST} \to \mathsf{SST}$$
$$\mathsf{Z}\;(X \otimes Y) = \mathsf{Z}\;X \times \mathsf{Z}\;Y$$
$$\mathsf{S}\;(X \otimes Y)\;\langle\, x\,,\,y\,\rangle = (\mathsf{S}\;X\;x)\;\otimes^{\mathsf{d}}\;(\mathsf{S}\;Y\;y)$$

Displayed Coinductive Types

In dTT, SST is a special case of a *displayed coinductive types*

Here are some more examples of what we can do:

Displayed Coinductive Types

In dTT, SST is a special case of a *displayed coinductive types*

Here are some more examples of what we can do:

$$\textbf{codata}\ \mathsf{Pt}\ (X : \mathsf{SST}) : \mathsf{Type}\ \textbf{where}$$
$$\mathsf{zp} : \mathsf{Pt}\ X \to \mathsf{Z}\ X$$
$$\mathsf{sp} : (p : \mathsf{Pt}\ X) \to \mathsf{Pt}^{\mathsf{d}}\ X\ (\mathsf{S}\ X\ (\mathsf{zp}\ p))\ p$$

Displayed Coinductive Types

In dTT, SST is a special case of a *displayed coinductive types*

Here are some more examples of what we can do:

$$
\begin{aligned}
&\textbf{codata}\ \mathsf{Pt}\ (X : \mathsf{SST}) : \mathsf{Type}\ \textbf{where}\\
&\quad \mathsf{zp} : \mathsf{Pt}\ X \to \mathsf{Z}\ X\\
&\quad \mathsf{sp} : (p : \mathsf{Pt}\ X) \to \mathsf{Pt}^{\mathsf{d}}\ X\ (\mathsf{S}\ X\ (\mathsf{zp}\ p))\ p
\end{aligned}
$$

$$
\begin{aligned}
&\textbf{codata}\ \mathsf{Hom}\ (X\ Y : \mathsf{SST}) : \mathsf{Type}\ \textbf{where}\\
&\quad \mathsf{zhom} : \mathsf{Hom}\ X\ Y \to \mathsf{Z}\ X \to \mathsf{Z}\ Y\\
&\quad \mathsf{shom} : (f : \mathsf{Hom}\ X\ Y)\ (x : \mathsf{Z}\ X) \to\\
&\qquad \mathsf{Hom}^{\mathsf{d}}\ X\ (\mathsf{S}\ X\ x)\ Y\ (\mathsf{S}\ Y\ (\mathsf{zhom}\ f\ x))\ f
\end{aligned}
$$

Correctness of the Definition

We assume that our starting CwF for the discrete mode has ω-limits

This will be true for any ∞-topos model

Correctness of the Definition

We assume that our starting CwF for the discrete mode has ω-limits

This will be true for any ∞-topos model

Then we can show that SST is indeed a classifier for semi-simplicial diagrams

Correctness of the Definition

We assume that our starting CwF for the discrete mode has ω-limits

This will be true for any ∞-topos model

Then we can show that SST is indeed a classifier for semi-simplicial diagrams

On the other hand, we conjecture that there are models of dTT, perhaps obtained from realizability, that admit a construction of SST, in our sense, that do not admit a classifier for semi-simplicial diagrams

Correctness of the Definition

We assume that our starting CwF for the discrete mode has ω-limits

This will be true for any ∞-topos model

Then we can show that SST is indeed a classifier for semi-simplicial diagrams

On the other hand, we conjecture that there are models of dTT, perhaps obtained from realizability, that admit a construction of SST, in our sense, that do not admit a classifier for semi-simplicial diagrams

If this were the case, then our characterisation would be a more general notion of *internal* or *uniform* diagrams, as opposed to the *external* version presented at the start of this section

Thank you for listening to my talk!